

**Programmierung**  
**Prof. Dr.-Ing. Nobert Fuhr**

Gudrun Fischer  
Sascha Kriewel  
{fischer,kriewel}@is.informatik.uni-duisburg.de

## Übungsblatt Nr. 1

### Aufgabe 1: Compiler und Interpreter

Mache Dich mit Arbeitsumgebung und Editor vertraut.

Der Compiler, der aus dem Quelltext ausführbaren Programmcode herstellt, heißt bei der hier verwendeten Java-Variante von Sun *javac*. Er wird normalerweise mit dem Namen der zu kompilierenden Klassendatei und eventuell zusätzlichen Parametern aufgerufen, z.B.

```
javac ExampleClass.java
```

Zur Erinnerung: aus *SciTE* heraus kann man eine geöffnete Quelldatei mit **Ctrl-F7** kompilieren.

Tippe zunächst einmal das folgende kurze Programm ab. Es soll Dein erstes Java-Programm werden. Speichere es als **Hallo.java** ab (beachte dabei die Groß- und Kleinschreibung). Kompiliere das Programm und führe es aus.

Listing 1: Hallo.java

```
1 public class Hallo
2 {
3     public static void main (String [] args)
4     {
5         System.out.println ("Hallo_Welt!");
6     }
7 }
```

Ist ein Quelltext syntaktisch oder grammatikalisch fehlerhaft, so kann schon der Compiler Fehler finden. Das hilft Dir eventuell, wenn Du beim Abtippen des obigen Programmes Fehler gemacht hast.

Das folgende kurze Programm ist absichtlich leicht fehlerhaft. Tippe es zunächst ab und speichere es als **Summe.java**. Du kannst es probeweise auch unter einem anderen Namen speichern, und ausprobieren, wie sich der Compiler dann verhält.

Rufe anschließend den Compiler auf und versuche, das Programm durch Lesen der Fehlermeldungen zu „entlausen“, bis es sich erfolgreich kompilieren und mit F5 oder dem Befehl `java Summe` ausführen lässt.

Listing 2: Summe.java

```
1 public class Summe
2 {
3     public static main (String [] ARGS)
4     {
5         int summe
6         int summand1 = 12.5,
7         int summand2 = 24;
8
9         { summe = summand1 + summand2; }
10        ;
11        System.out.print (Summand1);
12        System.out.print ("");
13        System.out.print (Summand2);
14        System.out.print ('=');
15        System.out.println (summe);
16
17        double wert = 30.4;
18        int summand2 = int (wert);
19        summe = summand1 + summand2;
20
21        System . out . print (Noch eine Summe: );
22        System.out.println (summe);
23 }
```

### Aufgabe 2: Die „Java API“

API ist ein Akronym für **Application Program(ming) Interface**. Die API beschreibt die verfügbaren Sprachelemente, so dass ein Programmierer sie benutzen kann. Auf der Webseite der Vorlesung befindet sich eine Verknüpfung zur Java API in der Version 1.4.2. Öffne die Seite in einem Webbrowser und mache Dich mit dem Aufbau dieser Dokumentation vertraut.

Suche die „Klasse“ `System`, die in obigem Beispiel benutzt wird. Informiere Dich über `System`, das „Objekt“ `out` und den Unterschied zwischen den „Methoden“ `println` und `print`.

Wie muss man den Quelltext der Klasse `Summe` ändern, damit es bei der Bildschirmausgabe zwischen dem Gleichheitszeichen und der ersten Summe eine Leerzeile gibt?

### Aufgabe 3: Parameterübergabe

Beim Aufruf eines kompiliertes Java-Programms ist es möglich, Parameter mit zu übergeben. Tippe das folgende kurze Programm ab und speichere es als

`Test.java`. Kompiliere das Programm und führe es dann *nicht* mit F5 aus, sondern rufe es im Ausführungsfenster mit dem Befehl `java Test 1 2` auf.

Die übergebenen Parameter sind über das Zeichenkettenfeld `args` aus dem Programm heraus zugreifbar. Die Nummerierung erfolgt wie in der Informatik üblich bei 0 beginnend.

Listing 3: Test.java

```
1 public class Test
2 {
3     public static void main (String [] args)
4     {
5         System.out.println (args [0]);
6         System.out.println (args [1]);
7     }
8 }
```

Ändere nun das Programm `Hallo` so ab, dass man es mit

```
java Hallo "Dein Name"
```

aufrufen kann und es dann statt „Hallo Welt!“ den Gruß „Hallo Dein Name!“ ausgibt.